HANDWRITTEN BANGLA COMPOUND CHARACTER RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK

By

Kazi Rifat Ahmed

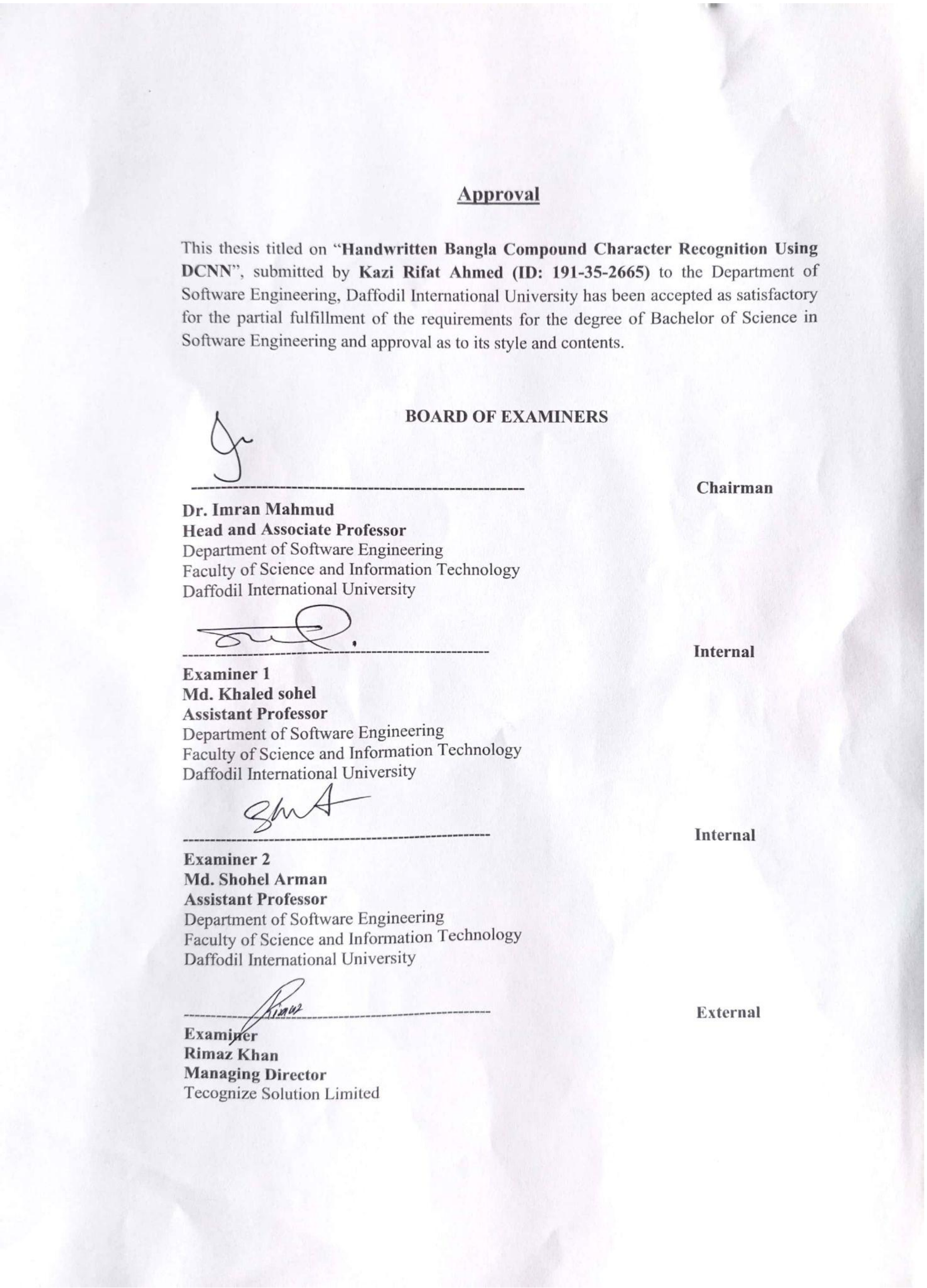
ID: 191-35-2665

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

FALL – 2022



Approval

This thesis titled on "**Handwritten Bangla Compound Character Recognition Using DCNN**", submitted by **Kazi Rifat Ahmed (ID: 191-35-2665)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Head and Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal

Examiner 1
Md. Khaled sohel
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal

Examiner 2
Md. Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External

Examiner
Rimaz Khan
Managing Director
Tecognize Solution Limited

DECLARATION

This is to clarify that the work presented in this thesis paper is the result of the investigation and research carried out by **Kazi Rifat Ahmed** under the supervision of **Ms. Nusrat Jahan**, Assistant Professor, Department of Software Engineering, Daffodil International University.

It is also declared that neither this thesis paper nor any part of the paper has been submitted anywhere else for the award of any degree, diploma, or other qualifications.

Supervisor:

Nusrat

____

Ms. Nusrat Jahan
Assistant Professor
Department of Software Engineering
Daffodil International University

Submitted by-

Rifat

____

Kazi Rifat Ahmed
ID: 191-35-2665
Department of Software Engineering
Daffodil International University

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Ms. Nusrat Jahan**, Assistant Professor, of the Department of Software Engineering, Daffodil International University, for her unwavering support, assistance, inspiration, and skillful direction during the course of her supervision. I have been able to work with extensive datasets and cutting-edge theories and methodologies in this field because of her abundant resources and practical ideas on the subject of Computer Vision and Deep Learning. She has always been an excellent mentor for a total beginner like me, and her praise and helpful criticism at each stage of my thesis has only strengthened and given me the confidence I need to successfully finish the dissertation.

I am appreciative of the support and departmental help I received from **Dr. Imran Mahmud**, Associate Professor and Head, Department of Software Engineering, Daffodil International University, throughout my B. Sc. In Software Engineering studies there.

Additionally, I want to thank all of my teachers and friends for their help and inspiration during the work process. I owe my parents for their encouragement and emotional support.

Finally, I want to express my gratitude to Almighty for blessing me with so many reputable people in my life in general and with this auspicious accomplishment in particular.

Kazi Rifat Ahmed
ID: 191-35-2665
Department of Software Engineering
Daffodil International University

CONTENTS

ABSTRACT

Optical character recognition for Bangla handwritten character is an important task for our daily life. Recognizing handwritten characters has difficulties because it differs from person to person. Thus recognizing handwriting characters can be very challenging. In this paper, I have introduced two different DCNN models for recognizing Bangla compound characters.In first model, I have used 6 convolution layer, 3 dropout layer and batch normalization in each layer with LeakyReLu activation function and other proposed model has 6 layers of convolution layers and 2 dropout layers, ReLU as an activation function to recognize 171 classes of Bangla handwritten character characters. The model was tested on the AIBangla dataset of compound characters. My proposed model was trained to recognize 171 characters using the DCNN model. Among those two, model 2 provided highest accuracy of 76.12%. There is still room for improvement, these results are significantly better than other models.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
OCR	Optical Character Recognition
MLP	Multilayer Perceptron
SVM	Support Vector Machine
ANN	Artificial Neural Network
HOG	Histogram of Oriented Gradient
LRF	Longest Run Feature
CH	Chain Code Histogram
GW	Gabor Wavelet
LBP	Local Binary Pattern
LGP	Local Gradient pattern
CENTRIST	CENsus TRansform hISTogram
NABP	Noise Adaptive Binary Pattern
LTP	Local Ternary Pattern
LAID	Local Adaptive Image Descriptor
DTCTH	Discriminative Ternary Census Transform Histogram
ELU	Exponential Linear Unit
RELU	Rectified Linear Unit
SUM	Support Vector Machine

CHAPTER 1

INTRODUCTION

1.1 Background

Bangla is an Indo-Aryan language native to the Bengali region of South Asia. Bangla language is the seventh most spoken language all over the world and around 300 million people speak the Bangla language. Bangla language is the official and national language of Bangladesh. About 98% of the population of Bangladesh uses the Bangla language on a daily basis. Bangla characters are sets of vowels and consonants as well as compound characters. Compound characters are a set of two or more basic characters combined together. There are more than 171 compound characters in the Bangla language.

Nowadays the recognition techniques of handwritten characters are attracting many researchers in this area. Bangla OCR is an important research area that analyzes the image of handwritten documents. The researchers are mainly focused on Bangla basic and numeric characters. Only a few research are done on Bangla compound characters. Recognizing Bangla compound characters is a difficult task because of the shape of each character and it also varies from person to person. There are many uses of optical character recognition in real life such as banking automation, postal office automation, form processing, signature verification, processing digital files, etc. There are different types of handwritten character recognition techniques. These methods are Multilayer Perceptron (MLP), Support Vector Machine (SVM), Artificial Neural Network (ANN), deep-learning-based models, and neural networks. Among these methods Convolutional Neural Network (CNN) based methods are gaining more usage in OCR research due to its better accuracy and low pre-processing. For image classification, CNN is a widely used technique nowadays. The major advantage of using CNN is that it automatically detects all the important features without any supervision. Some examples of Bangla characters can be seen in Table 1.1.

Table 1.1: Some Bangla basic and compound characters

Basic Characters		Compound Characters	
র	স	শ্ব	স্ব
ক	ঙ	স্ট	স্ল

1.1.1 A Brief on Bangla Character

Bangla language structure is quite different than any others European language. It consists of 50 basic characters where 11 vowel characters and 39 consonant characters. Again, there are compound characters that are the combination of two or more basic characters. By combining two or more characters we can create different variations of compound characters. There are about 171 or more Bangla compound characters. Bangla characters have a horizontal line at the upper part of the characters. That horizontal line is called “matra” or “line”. In a few of the characters, there are some exceptions. There is no "matra" in several of the Bangla characters. It will be incorrect if someone adds "Matra" to those characters. There are several examples of characters in Table 1.2 that don't include any "Matra" at all. Bangla characters always flow from left to right. Bangla characters are not classified as uppercase or lowercase like in other languages. In this paper, Bangla compound characters will be used for recognition.

Table 1.2: Few examples of Bangla characters without “matra”

ঙ	এ	ঋ

1.1.2 Characters of Bangla Compound Script

There are several Bangla compound characters. Among those Bangla scripts, Table 1.3. has shown 171 Bangla compound characters.

Table 1.3: Bangla compound characters

প্র	ঙ্গ	ক্ষ	ত্র	ন্দ	চ্ছ	ন্ত	ন্দ্র	স্ত	ক্ত	ম্ব	স্ট	গ্র	স্থ	স্ব
ত্ত	ন্ট	ল্প	ষ্ট	ক্র	ন্ন	দ্ধ	ন্ধ	ঙ্ক	ন্ড	ফ্র	ম্প	স্ক	জ্ঞ	ক্ট
শ্চ	ট্র	ত্ব	ল্ল	ব্র	ঞ্চ	ন্ড	ক্স	শ্র	দ্র	স্প	জ্ঞ	ন্ম	ম্ভ	শ্ব
শ্ন	ব্দ	ব্ল	প্প	প্ত	ক্ল	ষ্ট্র	দ্ব	ট্ট	গ্ল	ল্ট	ষ্ঠ	স্ত্র	প্ল	ষ্চ
স্ম	দ্দ	গ্ন	জ্ব	ষ্ক	ত্ম	ড্র	ম্ম	ন্ট	ম্প্র	প্ন	ম্ম	স্ফ	ল্দ	ন্ত্র
ন্ত্ত	ত্থ	জ্জ	ক্ষ্ম	ষ্ণ	ন্ত্র	ক্ক	ন্থ	ব্ব	ড্ড	ন্ট্র	ন্ঠ	প্ট	স্ত্ত	ধ্ব
হ্ণ	ভ্র	ল্ক	স্ল	হ্ল	ত্ন	ষ্ক্র	ঘ্র	দ্ভ	শ্ল	ব্ধ	ষ্ম	স্ক্র	ড়্গ	জ্জ্ব
শ্ম	দ্ম	ক্ব	ম্র	গ্ধ	ব্জ	স্ন	ন্দ্ব	ঙ্ঘ	খ্র	হ্ম	থ্ব	ল্ব	ম্ন	ঘ্ন
গ্ন	ক্ষ্ণ	গ্রু	চ্ছ্ব	ন্ন	ল্ম	স্ন	ম্ল	ষ্প্র	ঞ্ঝ	ম্ভ	ষ্প	ঙ্খ	স্প্র	ত্রু
গ্ব	থ্ব	ন্ব	হ্ন	দ্ব	দ্ঘ	ধ্র	হ্ল	গ্ম	ল্গ	স্থ	থ্র	ন্ধ্র	ফ্ল	ঙ্ক্ষ
ণ্ম	ঞ্ছ	ম্ফ	হ্র	প্রু	ক্রু	ভ্ল	শ্রু	দ্রু	ভ্ম	চ্ছ্র	ক্ম	দ্ন	ত্ত্র	ট্ব
প্স	ল্ড	ষ্ফ	শ্ছ	জ্ঝ	স্ট্র									

1.1.3 What is OCR

Optical character recognition is abbreviated by OCR. OCR is the mechanical or electrical process of converting any images of typed, handwritten, or printed text into machine-encoded text. OCR became popular in the early 1990s while digitalizing newspapers. Since then, OCR has gone into several improvements. OCR is mostly used to turn any hard copy of any document into PDFs or Doc files. So that other users can use that file to edit and format the way they want. It can be used in different ways such as recognizing text(license plates, letters, books, etc), translating an image into specific languages, passport recognition, traffic sign recognition, and many more.

The main advantages of OCR are that it saves time, decreased errors, and minimizes effort. OCR provides the functionality to be able to edit and search those documents. Figure 1.1 shows the process of OCR:

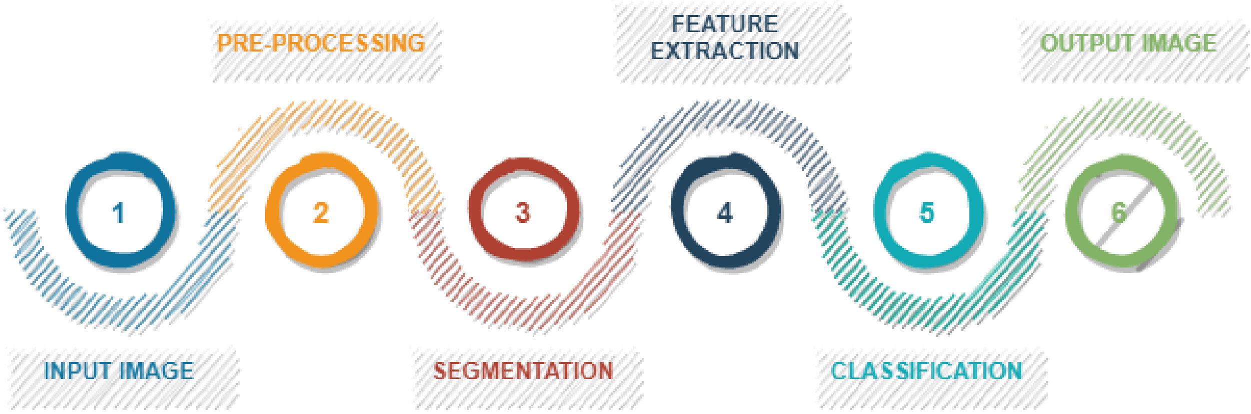


Figure 1.1: Process of Optical Character Recognition

1.1.4 Application of Bangla OCR

Bangla OCR has been successfully used in different applications. Some of the important applications are discussed below:

1. **Traffic Sign Recognition:** It is a system that can automatically read road signs and display them to drivers as they drive. This is useful if anyone is unfamiliar with the road and doesn't have a chance to look at the speed limit.
2. **Bank Automation:** It is typically utilized in banks to read checks. It's crucial to read bank checks since they deal with the security of a person's assets. As there are many fraud attempts, it is also used to confirm the account holder's signature.
3. **Passport Recognition:** It is used to scan the passport and photograph the passport's identification papers. It catches the distinctive illuminations to determine whether or not the passport is legit. It can assist someone avoids being a victim of identity theft.

4. **Postal office automation:** It is used to read handwritten postal addresses on mail. The handwritten postcode digits are read using the Bangla OCR system. It has the ability to understand this code and can do automatic mail sorting.
5. **Form Processing:** It can also be used to process forms. Forms are typically used to collect public data.
6. **License Plate Recognition:** It is used to read a license plate that has been caught on camera in a vehicle. It identifies the character on the license plate.

1.2 Motivation of the Research

Bangla compound character recognition is known for its difficulties. Many researchers have conducted research on Bangla basic characters and numeric characters. Though tremendous work has been done on Bangla characters, there are only a few works done on Bangla compound characters. I want to work on this part in order to improve the recognition of Bangla compound characters. To improve the recognition technique of compound characters CNN model will be used. Because Convolutional Neural Network (CNN) has the ability to recognize any kind of visual pattern from any pixelated images. CNN model has been used for image classification and character recognition in recent years. There are few CNN models being used to recognize Bangla compound characters with a lower accuracy rate. But using the DCNN model for Bangla compound characters will give satisfactory results in terms of recognition accuracy. In the DCNN model, the training phase requires much time but the testing phase takes less time. So, it can be applied to real-life applications. Therefore, I want to provide a DCNN model that will improve the accuracy of the other models that have already been suggested.

1.3 Problem Statement

Bangla character is divided into two categories: basic characters and compound characters. Basic characters have 11 vowels and 39 consonants. In the Bangla language, there are a large number of compound characters which are formed by combining two or more basic characters (Table 1.3). Most of the characters have a horizontal line at their upper parts, called “headline” or “matra” and some characters don't have any “headline” or “matra”. Compound characters are difficult to distinguish since handwritten characters differ from person to person. As a result, precisely identifying the character might become rather difficult.

(Hasan, et. al.,2019) created a dataset called the AIBangla dataset by collecting images from different people. Their dataset contains images of 50 classes of Bangla basic characters and 171 classes of Bangla compound characters. They have applied three state-of-the-art deep CNN models to their dataset. They want to expand their database in the future. In addition, I will use their dataset of compound characters in my research to develop a technique that will provide better results than previous models for identifying Bangla compound characters.

On Bangla basic characters, compound characters, and numerals, (Saha, et. al.,2018) examined several Local Binary Pattern (LBP)-based feature descriptors. The following features were evaluated and compared: LBP, LGP, CENTRIST, NABP, LTP, LAID, and DTCTH. For these, they used several datasets. Among these methods, DTCTH performed well in all three datasets, achieving accuracy rates of 99.2% for Bangla numbers, 93.24% for Bangla basic characters, and 73.7% for Bangla compound characters. DTCTH offered better accuracy than all other features combined. As we can see, their feature provides great results for Bangla's basic characters and numbers but lacks behind in compound characters. In the future, they want to apply deep neural network-based classification to improve the approximation. Therefore, my objective is to propose a DCNN model that will increase the accuracy to recognize those compound characters.

1.4 Research Question

This thesis research questions are as follows:

- How much accuracy does my model provide then other models in recognizing Bangla compound characters?
- Can deep learning models outperform traditional models?

1.5 Research Objective

The objectives of this thesis are given below:

- To develop an architecture of the DCNN model to recognize hand-written Bangla compound characters.
- To analyze the DCNN model and improve the recognition accuracy of hand-written Bangla compound characters.
- To evaluate the performance of the DCNN method with other existing methods in terms of accuracy on a publicly available dataset.

The outcome of the thesis is a recognition method for handwritten Bangla compound characters with minimal storage needs and processing time that would increase recognition accuracy.

1.6 Research scope

The research scopes of this thesis are as follows:

- Reduce the possibility of inaccurate handwritten Bangla compound character detection in the near future.
- For handwritten Bangla compound characters, it will offer a high accuracy rate.
- It is applicable to real-world applications such as bank insurance paperwork, car number plates, newspaper, and others.
- Regarding this field, it will make some beneficial ideas.

1.7 Thesis Organization

The thesis is organized as follows:

In chapter 1, I have briefly discussed the background of this paper, what is OCR, the application of OCR, the motivation of the research, the problem statement, research questions, objectives, and scopes.

In Chapter 2, I have discussed the literature review on how other researchers have done their works and what they proposed.

In Chapter 3, I have discussed research methodologies on how I complete my work like data collection, data preprocessing process, the proposed DCNN model, and their details.

Chapter 4 describes the experimental results and analyses of the proposed model which has model accuracy and loss, confusion matrix, and other results that are necessary for the paper.

Finally, Chapter 5 provides the findings and contributions with the scope for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In recent years, handwritten Bangla characters and digit recognition have received a lot of attention from researchers. Because of their distinctive form, recognizing Bangla characters can be difficult and handwritten Bangla characters might be far more difficult to read because everyone's writing style differs. To solve these challenges, researchers experimented with several ways for improving the identification of handwritten Bangla characters. Deep neural network-based solutions, particularly convolutional neural network (CNN)-based solutions, have recently demonstrated considerable accuracy in image classification, recognition, and a variety of other domains that learn, extract, and classify features automatically.

Numerous researchers have already conducted their studies and used various machine-learning techniques to recognize Bangla compound characters. So, for my work, I have focused on only recognizing compound characters.

2.2 Related Works

(Ahmed, et. al.,2019) suggested a DCNN model to recognize the Bangla alphabet. For that, they used 76000 images. Their DCNN alphabet recognition model had a 95% success rate. They intend to modify their model in the future to introduce new feature selection filters.

To increase accuracy, (Ashiquzzaman, et. al.,2017) implement the DCNN model with an ELU activation function and Dropout. They used the CMATERDB 3.1.3.3 dataset of compound characters. They used ELU and Dropout to reduce data overfitting and overcome the vanishing gradient problem. The proposed approach of enhancing the classification accuracy of handwritten Bangla compound characters had a 93.68% accuracy.

(Azad, et. al.,2020) introduce Autoencoder with DCNN which they called it DConvAENNet. An autoencoder is pursued to attempt to copy its input to its output. For feature extraction and classification, they used the ReLU, Sigmoid, and Softmax activation functions. They used three different types of datasets such as BanglaLekha-Isolated, CMATERdb 3.1, and Ekush. Their proposed DConvAENNet model achieved 95.21% on BanglaLekha-Isolated for 84 classes, 92.40% on CMATERdb 3.1 for 238 classes, and 95.53% on Ekush for 122 classes. Their long-term goal is to enhance the autoencoder such that it may be demonstrated in the field of HCR.

To recognize Bangla digits using four distinct architecture models, (Basri, et. al.,2020) used the NumtaDB dataset. AlexNet offered the highest accuracy rating of 99.01% among those models, compared to both the normal and augmented data. They improved the model's accuracy by using augmented data. They want to increase accuracy by putting a transfer learning approach in the future. The model was only used to identify numbers.

To identify Bangla handwritten characters, (Begum, et. al.,2018) performed feature extraction. They used the CH, LR, and GW types of feature extractions. A standard database of handwritten Bangla characters was used by Begum et al. The most accurate feature extraction, LR+CH, achieved 76.47% accuracy. They'll work on Bangla compound characters in the future.

(Bipu, et. al.,2018) proposed handwritten character recognition based on a feature fusion endeavor. For distinct features, they used the HOG and Gabor filters. Their technique was implemented on 12000 images. By combining, HOG and Gabor filters they got a benchmark accuracy of 96.1%. Their work was done on only Bangla basic characters. In the future, they will extend their model to work on Bangla compound characters.

(Chakraborty, et. al.,2021) proposed a method that could convert basic characters into compound characters and transfer compound characters to basic characters by integrating deep learning and image processing approaches. For this experiment, they used a dataset that contains more than 300,00 images. At the time of the experiment, 70% of the dataset was used as training data, and 30% was used as testing data which got an accuracy of 89.20%. In the future, They will use more handwritten images to get high performance.

(Chowdhury, et. al.,2019) used a Banglalekha-isolated dataset to recognize Bangla handwritten characters by applying the CNN model with augmented data. With CNN and CNN augmented data, they achieved an accuracy of 91.81% and 95.25%, respectively. Their future scope was to detect a sequence of characters. In handwritten Bangla characters, it demonstrates the distinction between implementing raw and augmented data.

For handwritten isolated Bangla compound characters, (Fardous, et. al.,2019) used the DCNN model with a ReLU activation function and Dropout to improve the accuracy rate. They used CMATERDB 3.1.3.3 dataset. The ReLU activation function was used to overcome the vanish gradient problem. The accuracy of the proposed method was 95.5% which was higher than other methods. In the future, further improvement and analysis will be used by different datasets.

(Hakim, et. al.,2019) applied the CNN architecture VGG-16 model for handwritten Bangla numerals and basic characters. They used a BanglaLekha-Isolated dataset and for cross-validation, they created a new dataset that contains 6000 images. For their strategy, they tested 3 different image sizes: 32x32, 64x64, and 80x80. Only the Bangla basic characters and digits were recognized by their methods, which had a 99.82% accuracy rate. They will improve their performance in the future by using a larger image size and a deeper model.

(Hasan, et. al.,2019) created this AIBangla dataset by collecting images from different people for the purpose of using the dataset for isolated basic and compound characters. They have applied three state-of-the-art deep CNN models to provide a baseline performance. They used VGG-16, ResNet-50, and DenseNet. Among these, Basic-50 Character Set accuracy for DenseNet was 98.13% and Compound-171 Character Set accuracy for DenseNet was 81.83%. They want to expand their database of images in the future.

(Hasan, et. al.,2021) used the DCNN model for an automated system for recognizing isolated handwritten Bangla characters. They used CMATERdb 3.1.2 dataset. Their model achieves 95% accuracy in recognizing the alphabet. In order to show how to develop a complete Bangla handwritten text recognition system in the future, they will further modify the model.

In this paper, (Kibria, et. al.,2020) proposed a technique that extracts features to classify the complex structural variations of Bangla compound characters. To recognize Bangla compound characters, they used three types of features: LRF, HOG, and Diagonal. They used CMATERdb 3.1.3.3 dataset for their method. As a result, the SVM classifier has a success rate of 88.73% while the MLP classifier has a success rate of 85.91%.

Using the SE-ResNeXt, (Khan, et. al.,2022) created a DCNN model. To solve the handwritten compound character identification of Bangla, the squeeze and excitation (SE) blocks are added to the already existing ResNeXt. The BanglaLekha-Isolated 2 dataset was utilized. There were only 24 compound characters in their dataset. Their model received a 99.82% accuracy rating on average. They intend to create a new complex dataset in the future to compare the model with other datasets.

(Mukherjee, et. al.,2020) used supervised learning and unsupervised learning to recognize online Bangla basic characters. They applied MLP, KNN, BayesNet, Naive Bayes, SVM, and Random Forest to supervised learning and they used K-means clustering for unsupervised learning. They used a database of 10,000 images. They achieved 97.22% for supervised learning (using MLP classifier) and 90.69% accuracy for unsupervised learning (using K-means clustering). They aimed to design a more powerful feature to minimize stroke recognition errors.

For image classification, (Pramanik, et. al.,2018) used a chain code histogram feature set with an MLP-based classifier and backpropagation learning. In their method, they used ICDAR 2013 Segmentation Dataset and Cmaterdb dataset (version 3.1.3.1 for compound characters and version 3.1.2 for basic characters). On compound characters, their suggested method produced an overall accuracy of 88.74%. They will continue to work with this methodology for other Indian languages in the future.

(Purkaystha, et. al.,2017) developed a DCNN model with two convolutional layers, followed by three densely connected layers and the final dense layer is a softmax layer. Their model was developed to recognize Bangla handwritten characters. A BanglaLekha-Isolated dataset was used. The accuracy of the test was 98.66% for digits, 94.99% for vowels, 91.60% for compound letters, 91.23% for alphabets, and 89.93% for all characters. The gap in their model is that there is a chance that the network would seriously overfit.

(Reza, et. al.,2019) have constructed a CNN architecture that trained the basic character and transfer the trained weights for the compound character classification. For this approach, they used the Ekush dataset. They utilized the model layer by layer. Each layer provided test errors. The test errors for basic character for 1st layer 8.76%, 2nd layer 8.53%, 3rd layer 9.22%, 4th layer 12.26%, 5th layer 15.17% and The test errors for digit for 1st layer 7.09%, 2nd layer 6.86%, 3rd layer 7.41%, 4th layer 10.18 %, 5th layer 16.82 %. Among them, the 5th layer provides with highest test errors. In the future, they will evaluate the proposed method on other available Bengali character datasets for further validation.

In this paper, (Saha, et. al.,2019) proposed a new DCNN model called BBCNet-15. They could only recognize basic Bangla characters using this model. As for the dataset, they used CMATERdb 3.1.2 dataset which contains 50 character classes including 39 consonants and 11 vowels. The validation accuracy for BBCNet-15 was 96.40%, and the training accuracy was 98.86%. Compared to other model, their model accuracy is higher. In order to obtain more accurate findings in the future, they must implement the model to compound characters and digits.

(Saha, et. al.,2018) studied different Local Binary Pattern (LBP) based feature descriptors on Bangla basic characters, compound characters, and digits. These features were LBP, LGP, CENTRIST, NABP, LTP, LAID, and DTCTH which were evaluated and compared with one another. They used CMATERdb 3.1.2, CMATERdb 3.1.3.1, and CMATERdb 3.1.1 datasets for their method. Among these methods, DTCTH performed well in all three datasets obtaining 93.24% accuracy for Bangla basic characters, 73.3% accuracy for Bangla compound characters, and 99.2% for Bangla digits. In the future, they might use deep neural network-based classification to increase the approximation.

To recognize Telugu characters, (Sarika, et. al.,2021) implemented a CNN-based architecture VGG-16 model using a dataset that has a maximum of 1600 Telugu characters. In their model, they used two different sorts of input sizes. Model 1 utilized an input size of (224,224,3), while Model 2 used an input size of (128,128,3). Model 1's accuracy was 92%, and Model 2's accuracy was 80.50%. Both models suffered losses of 80.60% and 40%. Their accuracy and loss differential is significantly different due to the input size.

(Sharif, et. al.,2018) integrated HOG features in the learning classification of a deep learning model. They developed a HOG-CNN hybrid model. They used CMATERDB 3.1.3.3 dataset for classification. They tested the model on 171 and 199 classes of the datasets. Their model got an accuracy rate of 92.75% and 92.57% on both classes of datasets. Their future scope was to incorporate multiple features into their model.

2.3 Conclusion

In those papers, everyone used different types of CNN models for image classification. To have a better performance, they applied feature extractions, augmentation, annotation, and other techniques. But most of the better performance was found on Bangla basic characters and digit recognition. But to recognize Bangla compound characters their model performance was poor. So, in my work, I tried to improve the Bangla compound recognition accuracy for better performance.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, some key principles related to the proposed method are defined which include data collection, and data preprocessing, after which the proposed method is defined in detail.

3.2 Data Collection

The dataset I will use in this proposed method is Hasan, et al., 2019 AIBangla dataset. AIBangla is a new benchmark image database for isolated handwritten Bangla characters with a detailed performance and usage baseline. Their Bengali character image dataset has been written by more than 2,000 unique writers from different institutes across Bangladesh.

3.2.1 Dataset

AIBangla dataset has a huge collection of Bangla characters which contain Bangla basic and compound characters. They gathered data for 50 classes of Bangla basic characters, which included 80,403 images, and 171 classes of Bangla compound characters, which included 249,911 images, for their entire dataset. There is no numerical data in the AIBangla dataset. They have thus gathered 330,314 images for 221 classes as their entire dataset. I'll be using their dataset of Bangla compound characters. I will be using 50,069 images of 171 classes of Bangla compound characters for this study. There are a few examples of Bangla compound characters given below in Figure 3.1:

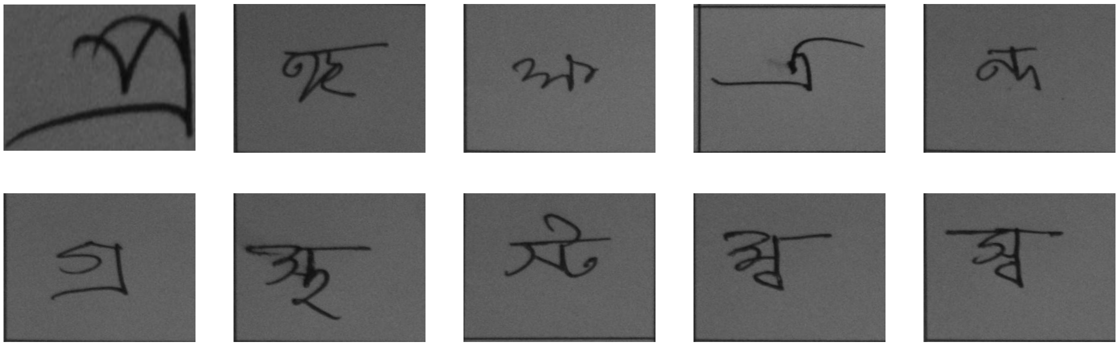


Figure 3.1: Few examples of Bangla compound characters of AIBangla dataset

3.3 Data Preprocessing

To build a deep learning or computer vision method, we always need data. In my case, I am using image data. But there are a few problems with image data that cause those data might have complexity, inaccuracy, and adequacy. So, to build a model it is essential to pre-process those data in order to achieve the desired results. I will be pre-processing those images, which will allow deep learning methods to solve them. Data pre-processing is used to increase the accuracy and also reduce its complexity. There are a few techniques used for preprocessing using image data. From Figure 3.2, those techniques are briefly discussed below:

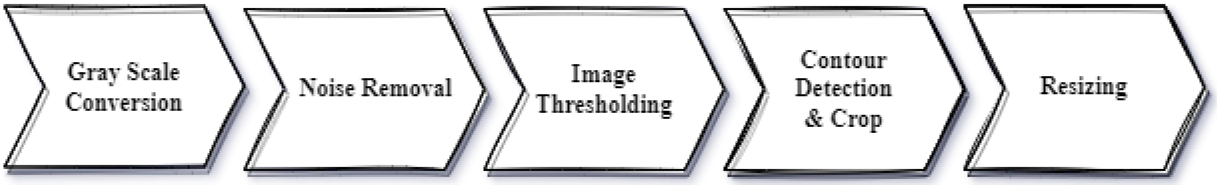


Figure 3.2: Image preprocessing steps

3.3.1 Gray Scale Conversion

Grayscaling is a process of converting an RGB image into a black-and-white image. RGB images are three dimensions. It helps to reduce the complexity of the model. AIBangla dataset has tone variation and noise in their dataset. So, it is important to convert those images into grayscale form. So that I can remove noise from the dataset. It will help to reduce the complexity of the model. I am using the python OpenCV library to convert those images into the grayscale format. Figure 3.3 shows the effect of applying grayscaling to an image.

Before grayscale conversion

After grayscale conversion

Figure 3.3: Example of grayscale conversion

3.3.2 Noise Removal

Noise removal is a process of removing or reducing the noise from an image. It reduces or removes the visibility of noise by smoothing out the entire image. To remove noise from the dataset images I have used the gaussian filter method. I have used the python OpenCV library to implement gaussian blur. It soothes the images of the dataset which helps to remove noises from images. The effect of applying a gaussian filter to the image is given in Figure 3.4:

Before noise remove

After noise removes

Figure 3.4: Example of noise removal from an image

3.3.3 Image Thresholding

Thresholding is one type of image segmentation, where we change the pixels of an image to make it easier to analyze. In image thresholding, we convert an image from color or grayscale into a binary image that is simply black and white. For my data preprocessing I will be using the multi-otsu threshold technique. Multi-Otsu thresholding is used to separate the pixels of an input image into several different classes, each one obtained according to the intensity of the gray levels within the image. By determining the number of classes, multi-otsu calculates several thresholds. I am using the python OpenCV library to use the multi-otsu thresholding to the AIBangla dataset. Figure 3.5 shows the results of using multi-otsu thresholding to the datasets.

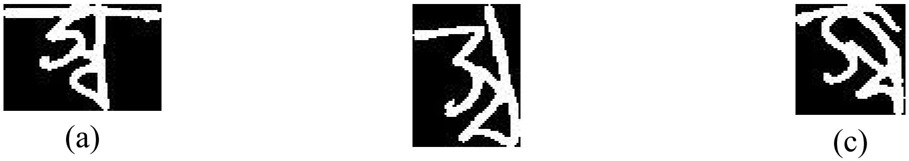
Before image thresholding

After image thresholding

Figure 3.5: Example of Multi-otsu thresholding techniques applied

3.3.4 Contours Detection and Crop

The unnecessary portion of the image must be removed in order to obtain the correct portion of the image that contains the handwritten compound characters. To put it another way, we can get the desired result by cropping the side of the images. I'm using a technique called contour detection to find the edges of Bangla compound characters. We can easily locate object borders in an image by using contour detection to identify their boundaries. To identify an image's contour, I'll use the Python OpenCV library. The image can be cropped to the exact size of the character after the image edge has been detected. A few examples of images with cropping and contour detection are shown in Figure 3.6.



(a)

(b)

(c)

Figure 3.6: (a), (b), and (c) are a few examples of contour detection and cropping

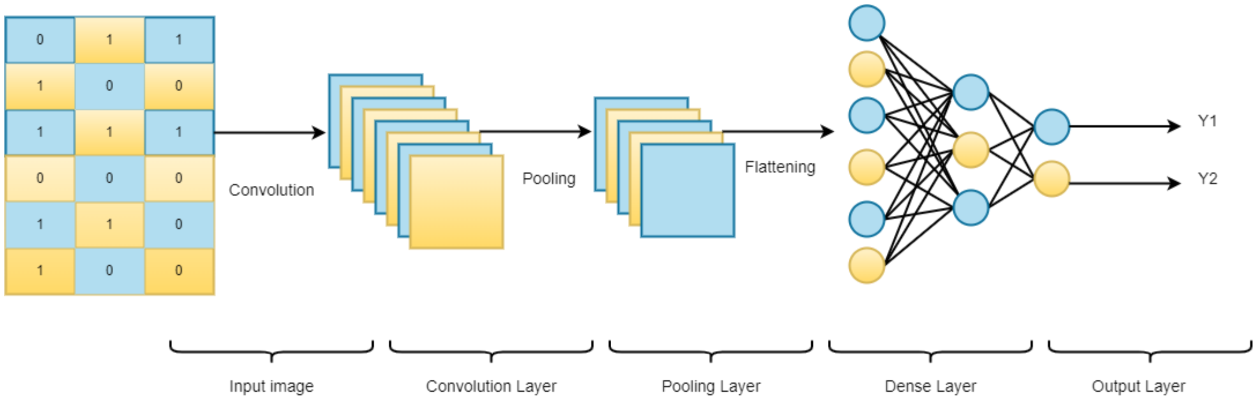
3.3.5 Image Resizing

Resizing images is an important step in image processing to achieve better results. A neural network can be trained faster by reducing the number of pixels in an image. I have resized my image to a 28 x 28 size to give my model a better result. The resized image of Bangla compound characters is shown in Figure 3.7.

Figure 3.7: Examples of 28 x 28 resized image

3.4 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning network architecture that learns directly from data. CNN is especially used for identifying patterns in images. Additionally, they are useful for identifying non-image data, such as audio, video, and other types. It is a kind of linear operation that enables the multiplication of two functions to produce a third function that can express how the shape of one function can be changed by the other. A CNN is constructed using many layers, including convolution, pooling, and fully connected layers, and it uses backpropagation to automatically learn data. An illustration of the CNN model is shown in Figure 3.8.



0	1	1
1	0	0
1	1	1
0	0	0
1	1	0
1	0	0

Figure 3.8: Convolutional Neural Network

3.4.1 CNN Structure

CNN typically has several layers. In a convolutional neural network, the following layers are frequently used:

3.4.1.1 Convolution Layer

Convolution layer is the base of a CNN model. They are responsible for carrying out convolution operations. The kernel/Filter executes the convolution operation. The kernel makes horizontal and vertical adjustments based on the stride rate until the entire image is scanned. The kernel is a 2-D array of weights that each represent a portion of the image. A dot product between the input pixels and the filter is calculated after the filter has been applied to a specific area of the image. The output array is next fed the dot product. Once the kernel has swept across the entire image, the filter shifts by one stride and repeats the process. The outcome of the series of dot products is called a feature map. Figure 3.9 has an example of a convolution layer.

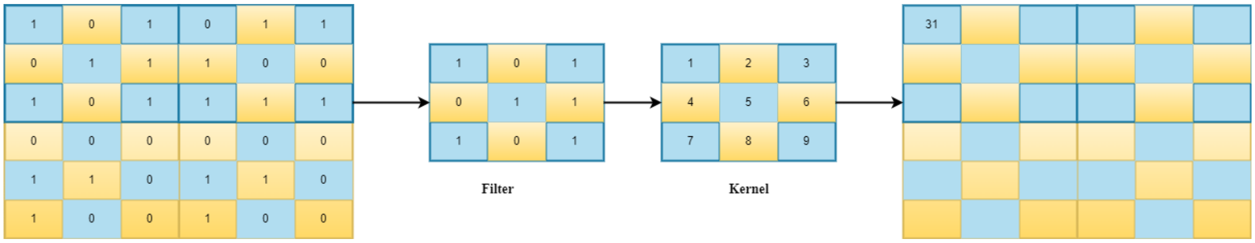


Figure 3.9: Example of a Convolution Layer

3.4.1.2 Pooling Layer

The pooling layer reduce the dimensionality of an image. It reduces the amount of computing power that is required to process the data. There are two types of pooling layers maximum pooling and average pooling. The maximum pooling layer returns the max value from the area of an image covered by the kernel and the average pooling layer return the average value of an image that is covered by the kernel. Figure 3.10 shows an example of a pooling layer.

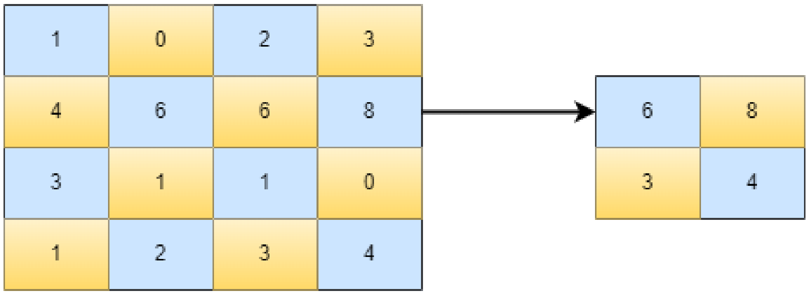


Figure 3.10: Pooling layer

3.4.1.3 Activation Function

An activation function is used to decide if a neuron should be activated or not. It will decide if the neuron's input to the network is important or not. There are several activation functions such as Sigmoid, Softmax, tanH, ReLU, Leaky ReLu, and more. The activation function is mostly used for multi-classification problems.

3.4.1.4 Dropout Layer

The dropout layer is used to nullify some neurons to the next layer while leaving other layers unchanged. It is also applied to hidden layers to nullify some hidden neurons. The dropout layer is important for CNN because they prevent the training data from overfitting. Figure 3.11 shows an example of a dropout layer.

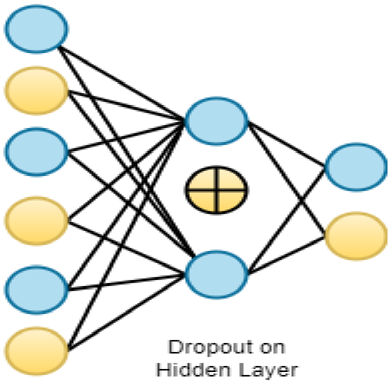


Figure 3.11: Dropout Layer

3.4.1.5 Fully Connected Layer

A fully connected layer has weights and biases which are used to connect the neurons between two layers. These layers make up the final few layers of a CNN architecture and are typically positioned before the output layer.

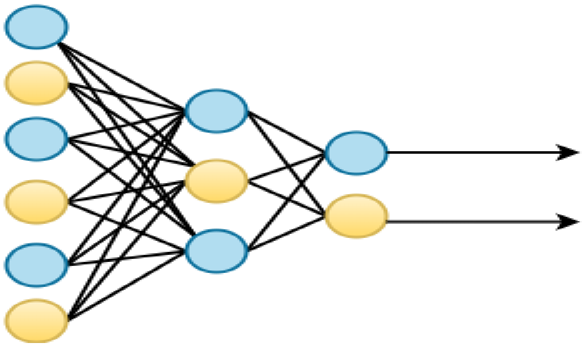


Figure 3.12: Fully Connected Layer

3.4.1.6 Batch Normalization

Normalization is a data pre-processing technique that is used to scale down the numerical data without changing its shape. Batch normalization transformation maintains the mean output and standard deviation output close to 0 and 1. It makes neural networks faster and more stable by adding extra layers in a deep neural network.

3.4.2 Proposed Method

The DCNN models used in this experiment are presented in this chapter. The performances of the DCNN structures that were tested for this thesis are shown. This chapter presents the architecture that provides the best results in terms of recognition accuracy rates.

3.4.2.1 DCNN Architecture

3.4.2.1.1 Proposed Model 1

This DCNN model consists of 6 convolutional layers and 3 Fully connected layers, and there are 2 dropout layers. It takes 28 × 28 images as input. 1st, 2nd, and 3rd convolution layers have 16, 32, and 32 numbers of filters. The size of the kernel is 3 × 3 in every layer. In each layer, there is a BatchNormalization layer. The pooling size for this model is 2 × 2 and this mode does not have any strides value. The LeakyReLU function is used as the activation function in each convolution layer. The max pooling layer is not used in 1st layer. After 3rd layer, the max pooling layer is called. There is a dropout layer after that to remove the overfitting problem.

In the 5th, 6th, and 7th convolution layers, the number of filters is 64, 64, and 128. Each layer contains an activation function. The LeakyReLU is used as an activation function. After the 6th layer, there is a max-pooling layer. In each layer, there is a BatchNormalization layer except the output layer.

The flatten layer is used after that. In the end, there are 3 fully connected layers which have 256, 200, and 171 number of neurons that are used as classifiers. In between every fully connected layer, there is a dropout layer. The last fully connected layer or output layer has a softmax activation function. The proposed DCNN model is shown in Figure 3.13.

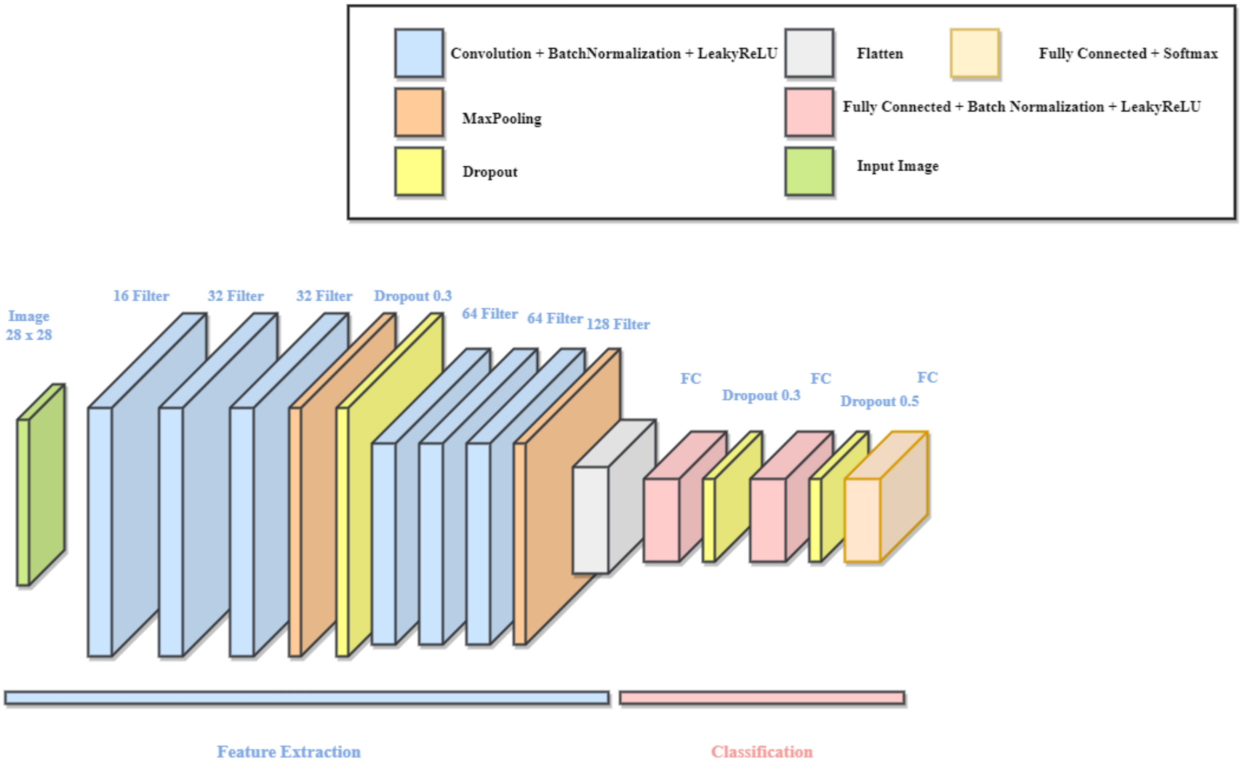


Figure 3.13: Proposed DCNN model 1

3.4.2.1.1 Proposed Model 2

This DCNN model consists of 6 convolutional layers and 2 Fully connected layers, and there are 2 dropout layers. It takes 28 × 28 images as input. In the input layer, I have rescaled all the images. 1st and 2nd convolution layers have 16 and 32 numbers of filters. The size of the kernel is 3 × 3 in every layer. The pooling size for this model is 2 × 2 and this mode does not have any strides value. The ReLU function is used as the activation function in each convolution layer. The max pooling layer is not used in 1st layer. After 2nd layer, the max pooling layer is called.

In the 3rd and 4th convolution layers, the number of filters is 32 and 64. Each layer contains an activation function. The ReLU is used as an activation function. After the 4th layer, there is a max-pooling layer.

Then, there are the 5th and 6th convolution layers which has 64 and 128 numbers of filters. At the 7th layer, the dropout layer was used, and then the max pooling layer is called. The flatten layer is used after that. In the end, there are 2 fully connected layers which have 128, and 171 number of neurons that are used as classifiers. In between the fully connected layer, there is a dropout layer. The last fully connected layer or output layer has a softmax activation function. The proposed DCNN model is shown in Figure 3.14.

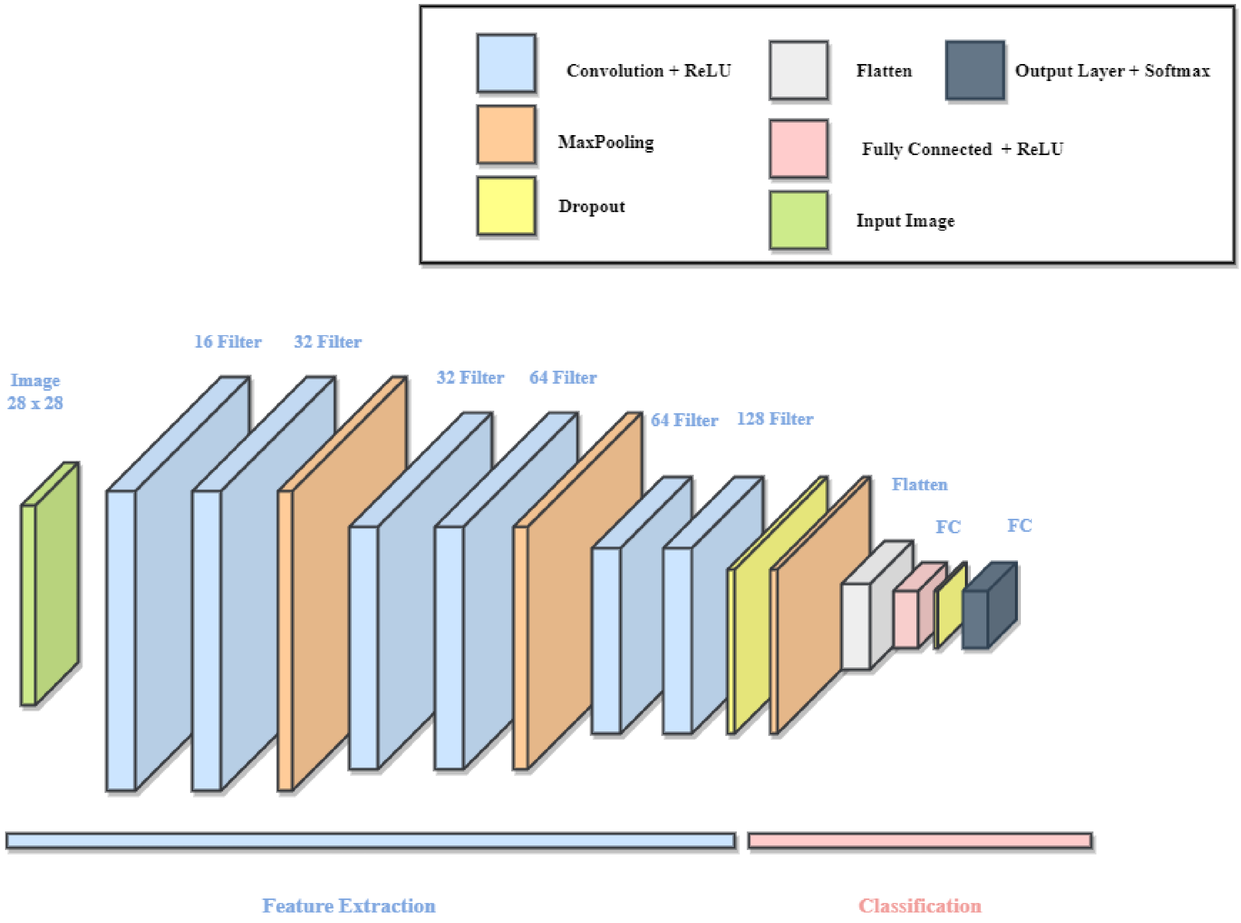


Figure 3.14: Proposed DCNN model 2

3.4.2.2 Diagram of the DCNN Model

Diagram of the DCNN model shows how many convolution layers, maxpool layer, dropout layer, dense layer and others layer that are necessary for a DCNN model are represented here.

3.4.2.2.1 Diagram of Proposed Model 1

In this Figure 3.15, the DCNN proposed model 1 is given. It shows what the models look like.

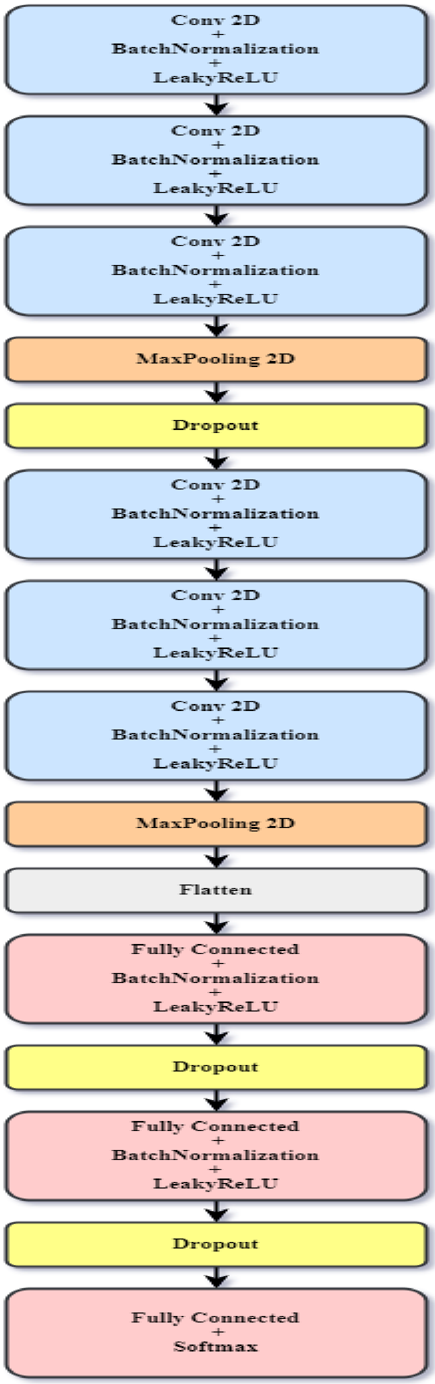

Conv 2D + BatchNormalization + LeakyReLU
Conv 2D + BatchNormalization + LeakyReLU
Conv 2D + BatchNormalization + LeakyReLU
MaxPooling 2D
Dropout
Conv 2D + BatchNormalization + LeakyReLU
Conv 2D + BatchNormalization + LeakyReLU
Conv 2D + BatchNormalization + LeakyReLU
MaxPooling 2D
Flatten
Fully Connected + BatchNormalization + LeakyReLU
Dropout
Fully Connected + BatchNormalization + LeakyReLU
Dropout
Fully Connected + Softmax

Figure 3.15: Diagram of DCNN proposed model 1

3.4.2.2.1 Diagram of Proposed Model 2

In this Figure 3.16, the DCNN proposed model 2 is given. It shows what the models look like.

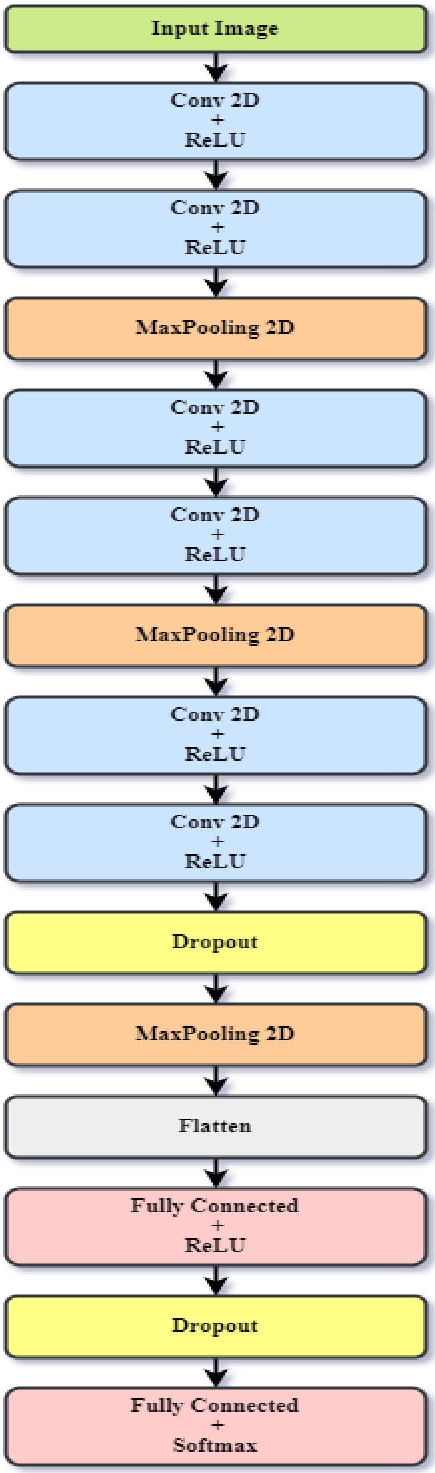


Figure 3.16: Diagram of DCNN proposed model 2

3.4.2.3 Details of the DCNN Model

3.4.2.3.1 Details of the Proposed Model 1

Table 3.1: Details of proposed model 1

Layer (type)	Output Shape	Parameter
Input Layer	28, 28, 3	
Conv2D	26, 26, 16	448
BatchNormalization	26, 26, 64	64
LeakyReLU	26, 26, 16	0
Conv2D	24, 24, 32	4640
BatchNormalization	24, 24, 32	128
LeakyReLU	24, 24, 32	0
Conv2D	22, 22, 32	9248
BatchNormalization	22, 22, 32	128
LeakyReLU	22, 22, 32	0
MaxPooling2D	11, 11, 32	0
Dropout	11, 11, 32	0
Conv2D	9, 9, 64	18496
BatchNormalization	9, 9, 64	256
LeakyReLU	9, 9, 64	0
Conv2D	7, 7, 64	36928
BatchNormalization	7, 7, 64	256
LeakyReLU	7, 7, 64	0
Conv2D	5, 5, 128	73856
BatchNormalization	5, 5, 128	512
LeakyReLU	5, 5, 128	0
MaxPooling2D	2, 2, 128	0
Flatten	512	0
Dense	256	131328
Dropout	256	0
BatchNormalization	256	1204
LeakyReLu	256	0
Dense	200	51400
Dropout	256	0
BatchNormalization	256	800
LeakyReLu	256	0
Dense	171	34371

Total params: 363,883

Trainable params: 362,299

Non-trainable params: 1584

3.4.2.3.2 Details of the Proposed Model 2

Table 3.2: Details of proposed model 2

Layer (type)	Output Shape	Parameter
Input Layer	28, 28, 3	
Conv2D	28, 28, 16	448
ReLU	28, 28, 16	0
Conv2D	28, 28, 32	4640
ReLU	28, 28, 32	0
MaxPooling2D	14, 14, 32	0
Conv2D	14, 14, 32	9248
ReLU	14, 14, 32	0
Conv2D	14, 14, 64	18496
ReLU	14, 14, 64	0
MaxPooling2D	7, 7, 64	0
Conv2D	7, 7, 64	36928
ReLU	7, 7, 64	0
Conv2D	7, 7, 128	18496
ReLU	7, 7, 128	0
Dropout	7, 7, 128	0
MaxPooling2D	7, 7, 128	0
Flatten	1152	0
Dense	128	147584
ReLu	128	0
Dropout	128	0
Dense	171	22059
Softmax	171	0

Total params: 313, 259

Trainable params: 313, 259

Non-trainable params: 0

3.5 Conclusion

The suggested DCNN model is provided in this chapter with a brief explanation. We can see all the details about those two proposed DCNN models. The results will be described and discussed in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter discusses the study's experimental findings. The problem is defined first. Following that, the experimental platform's hardware and software are discussed. At the end of the chapter, the performance of the proposed models is analyzed and evaluated.

4.2 Research Platform

For experiment purposes, the devices that I have used and their specification are described below. The proposed model experiment was conducted on a desktop machine (CPU: Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz, RAM: 8.00 GB, GPU: Intel(R) UHD Graphics, Hard Disk Drive: TEAM S510S M80 256GB SSD) in Windows 11 Pro 64-bit(10.0, Build 22000) environment. The algorithm ran on the Anaconda 2.3.1version platform with Jupyter Notebook version 6.4.12.

4.3 Result Discussion

After applying the DCNN models to the dataset, I have obtained the model's accuracy and loss plotting graph. For this experiment, I have used 80% of data for training and 20% data for validation for both models. The graph of the model's accuracy and loss is obtained by comparing the training accuracy with the validation accuracy.

4.3.1 Proposed Model 1

In a model, accuracy measures how well our model has predicted any data by comparing it with the true values. Figure 4.1 shows the model accuracy per epoch based on the training and validation data. It shows the accuracy of training and validation accuracy for 500 epochs.

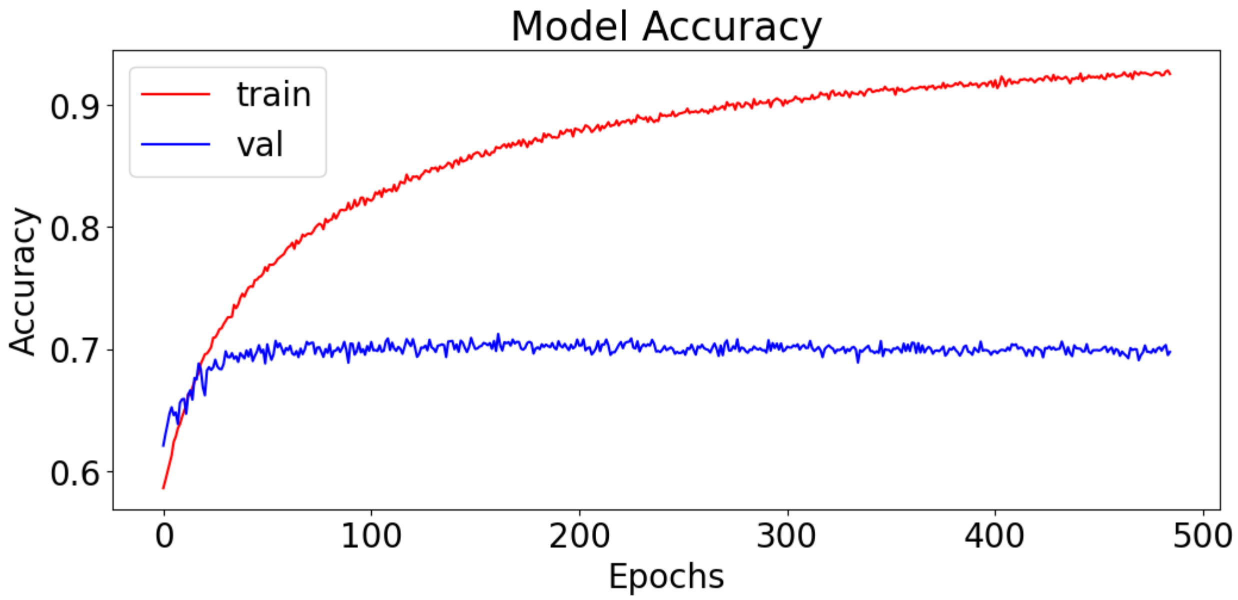


Figure 4.1: Model accuracy for proposed model 1

In a model, loss shows the number of errors. It measures how a model has performed. In a model, if the errors are high, that means the model does not return good results. Figure 4.2 shows our model loss based on model training and validation per epoch.

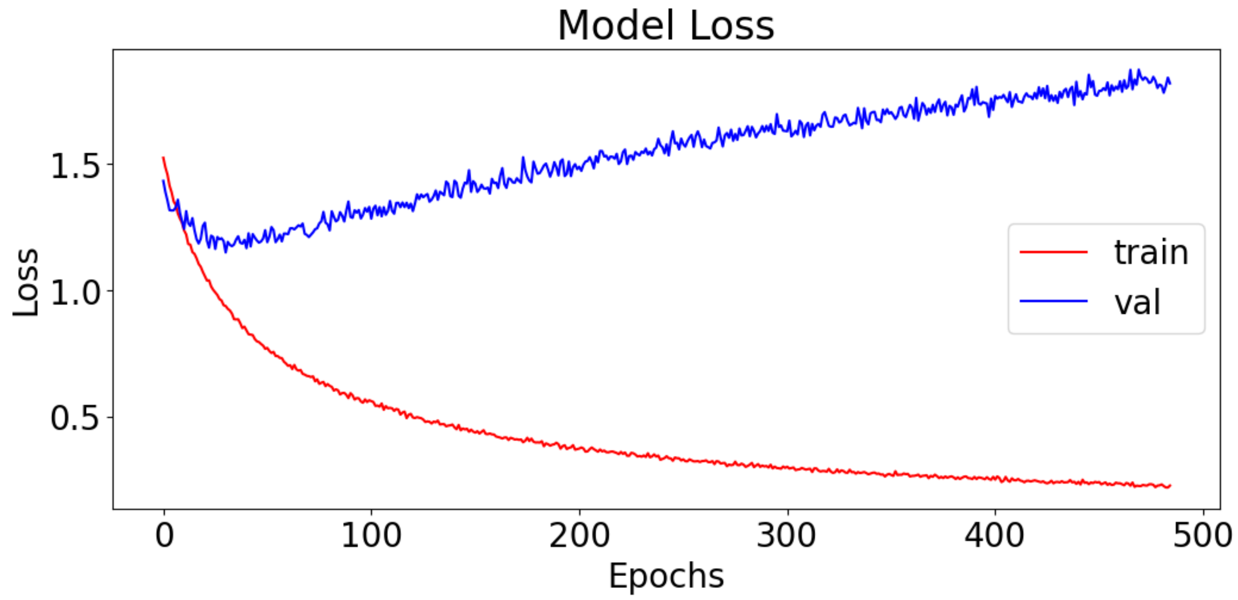


Figure 4.2: Model loss for proposed model 1

4.3.2 Proposed Model 2

In a model, accuracy measures how well our model has predicted any data by comparing it with the true values. Figure 4.3 shows the model accuracy per epoch based on the training and validation data. It shows the accuracy of training and validation accuracy for 100 epochs.

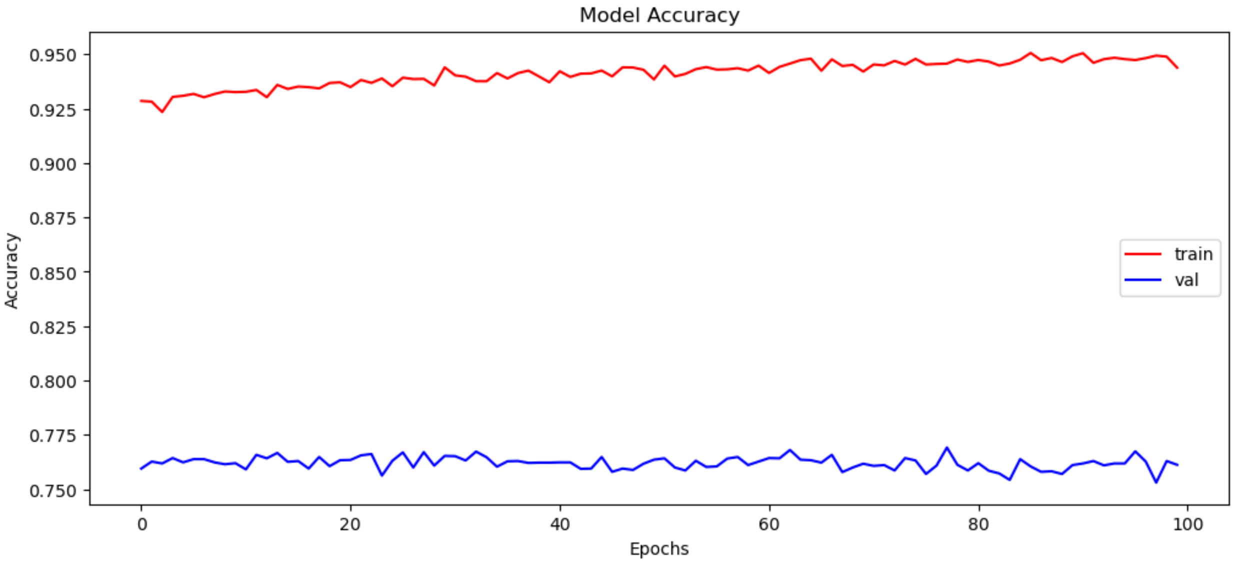


Figure 4.3: Model accuracy for proposed model 2

In a model, loss shows the number of errors. It measures how a model has performed. In a model, if the errors are high, that means the model does not return good results. Figure 4.4 shows our model loss based on model training and validation per epoch.

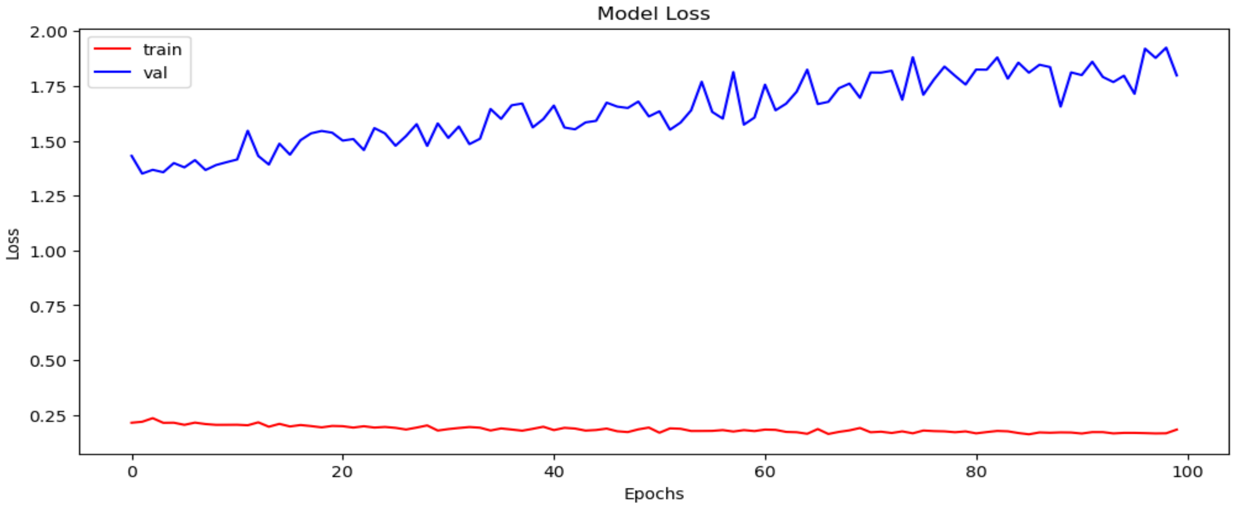


Figure 4.4: Model loss for proposed model 2

4.3.2.1 Performance Evaluation:

The accuracy of the DCNN for proposed models 1 and 2 is shown in table 4.1 after 500 training iterations for proposed model 1 and 150 training iterations for proposed model 2. The proposed DCNN model 1 obtains 69.34% recognition accuracy on the test dataset after 500 epochs and the proposed DCNN model 2 obtains 76.12% recognition accuracy on the test dataset and 76.122% recognition accuracy on the training dataset after 150 epochs. We can see the difference between their accuracy even after their various number of iterations. Model 1 iteration is higher than model 2 but the result has a huge difference. Model 2 provided better results than model 1. Model accuracy is determined by calculating the correct prediction of test data. Accuracy determines how well my model can predict from test datasets.

Table 4.1: Accuracy on DCNN proposed model 1 & 2

Model	Epocs	Train Accuracy	Test Accuracy
Proposed Model 1	500	69.34%	69.35%
Proposed Model 2	150	76.122%	76.12%

Confusion matrix defines how well any classification method work. It shows the actual value and predicted value in a matrix-like table form. Figure 4.5 provides the confusion matrix for the test samples for 171 classes of Bangla compound characters for the proposed model 1 and figure 4.6 provides the confusion matrix for the proposed model 2. The confusion matrix shows how many characters it predicted correctly. The number of samples and recognition accuracy for each class are shown in that figure.

In this chapter, figure 4.5 shows the confusion matrix for the proposed DCNN model 1.

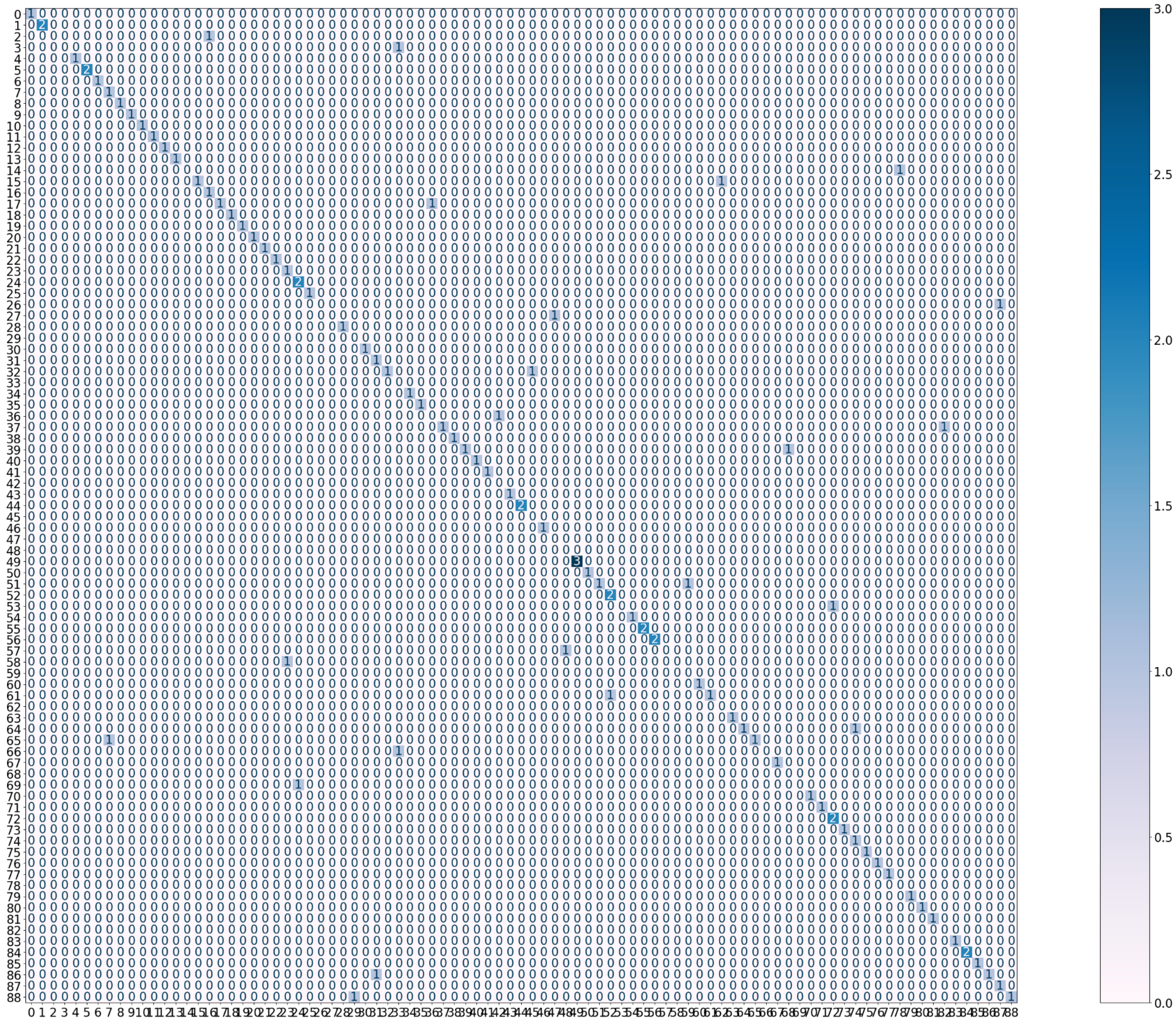


Figure 4.5: Confusion matrix for proposed model 1

In here, figure 4.6 shows the confusion matrix for the proposed DCNN model 2.

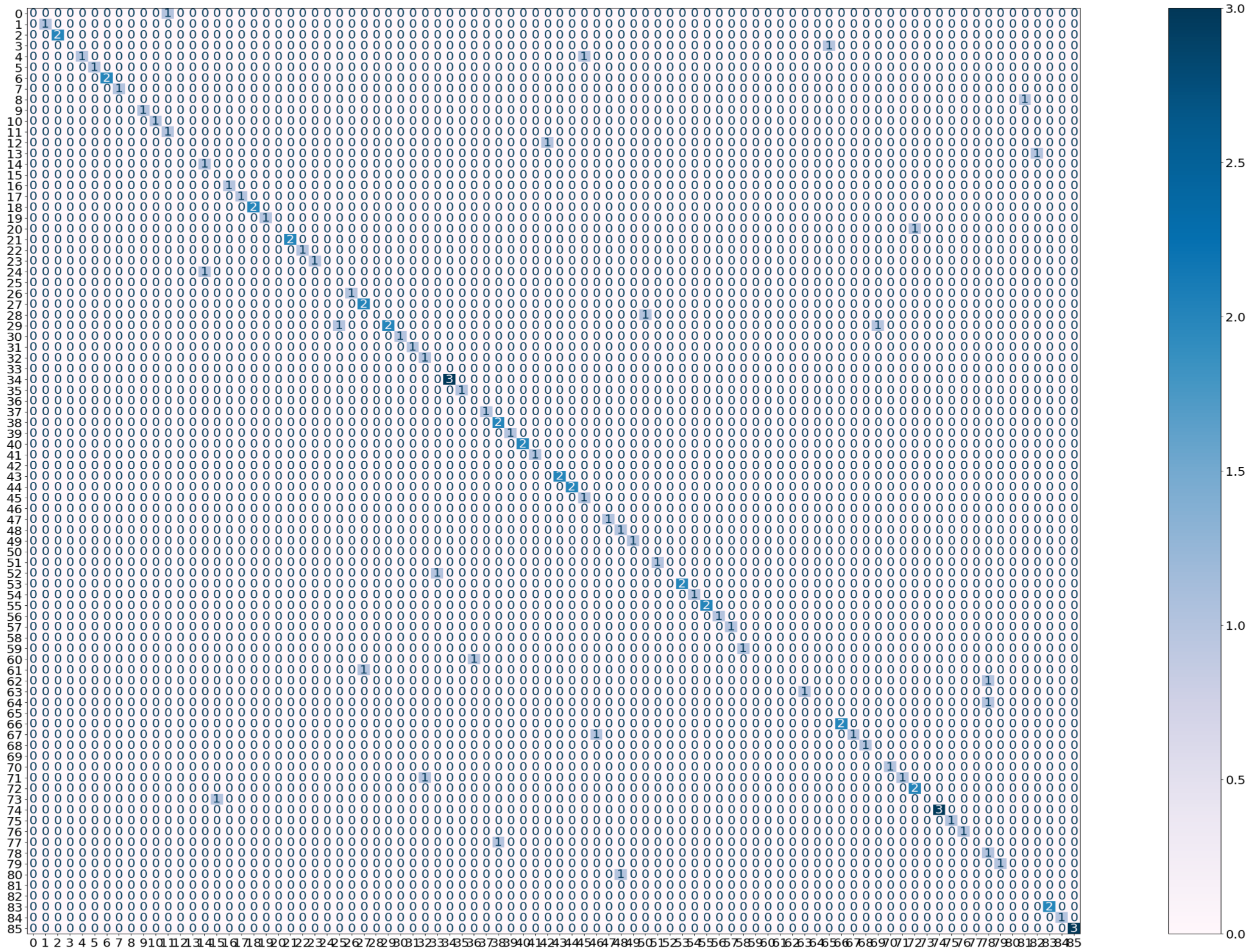


Figure 4.6: Confusion matrix for proposed model 2

4.3.3 Classification Results of Models

In Table 4.2 has the classification report of the proposed model which I have implemented for the classification method to recognize the Bangla compound characters.

Precision: The ratio of true positives to the total of true positives and false positives is called precision. A useful metric, precision reveals how accurate the prediction was among those considered to be positive.

$$\text{Precision} = \frac{True\ Positive}{True\ Positives + False\ Positives}$$

Recall: The ratio of true positives to the total of false positives and false negatives is called precision.

$$\text{Recall} = \frac{True\ Positives}{True\ positives + False\ Negatives}$$

F1-Score: F1-score is calculated using precision and recall.

$$\text{F1-Score} = 2\ * \ \frac{Precision * Recall}{Precision + Recall}$$

Accuracy: Accuracy is known as the ration between the correct prediction out of all other prediction made by a model. Accuracy help to determine any model's accuracy.

$$\text{Accuracy} = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + True\ Negatives + False\ Negatives}$$

Table 4.2: Classification results of proposed model

Model	Precision	Recall	F1-score	Support
Proposed Model 1	78%	78%	76%	100%
Proposed Model 2	78%	78%	76%	100%

As far as we can see, I have proposed two different types of DCNN model with different activation and batch normalization and filters. For my proposed model, model 1 provides precision at 78%, recall at 78%, and f1-score at 76%, and my second proposed model 2 provides precision at 78%, recall at 78%, and f1-score at 76%. My proposed model 1 provides 69.35% accuracy and my proposed model 2 provides an accuracy of 76.12% which give better accuracy results than the proposed model 1. So far, I can say, my proposed model 2 provides better performance than other models.

4.3.4 Models Accuracy

Table 4.3: shows comparisons between the proposed DCNN model and other proposed models to recognize Bangla compound characters.

Table 4.3: Comparison between Proposed model and other models

The work reference	Dataset	Feature Selection	Classification	Recognition Accuracy
(Saha, et. al.,2018)	CMATERdb 3.1.2, CMATERdb 3.1.3.1 and CMATERdb 3.1.1 dataset	LBP	SVM	65.69%
		LGP		63.17%
		CENTRIST		65.78%
		NABP		72.11%
		LTP		72.84%
		LAID		73.3%
		DTCTH		73.78%
Proposed Model 1	AIBangla	No	DCNN	69.35%
Proposed Model 2	AIBangla	No	DCNN	76.12%

As we can see that the proposed model 2 accuracy is better than other previously generated models. It provides a high accuracy than other existing models.

4.4 Conclusion

This chapter shows various aspects of my proposed model by highlighting different aspects. The proposed DCNN has been trained on 171 classes of Bangla character images. All of the results and demonstrations show that the suggested methodology can outperform other current ways of recognizing Bangla compound characters.

CHAPTER 5

CONCLUSION & RECOMMENDATIONS

5.1 Findings and Contributions

In recent years, CNN gain a lot of attention due to its better performance in image classification. Because it can train any model without any human supervision. The trained model can be applied at any moment. As a result, I applied the DCNN model to detect Bangla compound characters, which can help me obtain greater accuracy than the previously developed model. In this work, I have used a DCNN method to make better results. I have proposed 2 types of DCNN models in which model 2 provides better results than proposed model 1. My proposed model 1 has 6 convolution layers and 3 dropout layers and each layer has a batch normalization layer also I have used LeakyReLu as my activation layer and the proposed model 2 has 6 convolution layers and 3 max-pooling layers and 2 dropouts, and a dense layer. The activation function was ReLU and Softmax. I have used dropout to reduce the overfitting problem from the model to get better accuracy. So, based on my model, we can determine the value of recognition that provides the higher accuracy for our Bangla compound character. We validate the results by inspecting the automatically generated graphs. On each cycle, the accuracy and loss function values were used to construct the graphs. After implementing both proposed models I got an accuracy of 69.35% and 76.12%. In those two models, model 2 provides with higher accuracy rate. This model 2 is better than other models shown in Table 4.3.

My motivation was to propose a DCNN method that will improve accuracy for recognizing Bangla compound characters. (Saha, et. al.,2018) used different feature selections and SVM classifications for their model to recognize Bangla basic, compound characters, and digits. Among them, their compound character's recognition accuracy was very low. They wanted to implement a DCNN model to improve accuracy. So, I have proposed a DCNN model that improves the accuracy of the compound characters by achieving higher and better accuracy than their model. By using my model there is a chance to get better accuracy for other characters like Bangla basic characters and numeric characters.

5.2 Recommendations for Future Work

The model accuracy was not very satisfactory in this work. But this work has enormous potential for future expansion. Using better devices than mine can also help to improve model accuracy. It will save time and increase the potentiality for training the dataset.

The model accuracy can also be improved by using larger datasets. This proposed model was worked on a small dataset. So, there is bigger chance to improve it. By increasing the input size of the images to train the model can also help to get better results. This proposed model only works on single Bangla compound characters. In the future, I will try to use larger datasets and also try to recognize Bangla compound characters from any sentences.

REFERENCES

Ahmed, S., Tabsun, F., Reyadh, A. S., Shaafi, A. I., & Shah, F. M. (2019, July). Bengali handwritten alphabet recognition using deep convolutional neural network. In *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)* (pp. 1-4). IEEE.

Ashiquzzaman, A., Tushar, A. K., Dutta, S., & Mohsin, F. (2017, November). An efficient method for improving classification accuracy of handwritten Bangla compound characters using DCNN with dropout and ELU. In *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)* (pp. 147-152). IEEE.

Azad, M. A., Singha, H. S., & Nahid, M. M. H. (2020, November). Bangla handwritten character recognition using deep convolutional autoencoder neural network. In *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)* (pp. 295-300). IEEE.

Basri, R., Haque, M.R., Akter, M. and Uddin, M.S., 2020, January. Bangla handwritten digit recognition using deep convolutional neural network. In *Proceedings of the international conference on computing advancements* (pp. 1-7).

Begum, H., Rafid, A., & Islam, M. M. (2018, November). Recognition of Bangla Handwritten Characters using Feature Combinations. In *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-5). IEEE.

Bipu, M. M. C., & Afroge, S. (2018, February). A feature fusion based approach for handwritten Bangla character recognition using extreme learning machine. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)* (pp. 1-4). IEEE.

Chakraborty, S., & Paul, S. (2021, January). Bengali handwritten character transformation: basic to compound and compound to basic using convolutional neural network. In *2021 2nd international conference on robotics, electrical and signal processing techniques (ICREST)* (pp. 142-146). IEEE.

Chowdhury, R. R., Hossain, M. S., ul Islam, R., Andersson, K., & Hossain, S. (2019, May). Bangla handwritten character recognition using convolutional neural network with data augmentation. In *2019 Joint 8th international conference on informatics, electronics & vision (ICIEV) and 2019 3rd international conference on imaging, vision & pattern recognition (icIVPR)* (pp. 318-323). IEEE.

Fardous, A., & Afroge, S. (2019, February). Handwritten isolated Bangla compound character recognition. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-5). IEEE.

Hakim, S. A. (2019, February). Handwritten bangla numeral and basic character recognition using deep convolutional neural network. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-6). IEEE.

Hasan, M. M., Abir, M. M., Ibrahim, M., Sayem, M., & Abdullah, S. (2019, September). Aibangla: A benchmark dataset for isolated bangla handwritten basic and compound character recognition. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-6). IEEE.

Hasan, M. N., Sultan, R. I., & Kasedullah, M. (2021, April). An Automated System for Recognizing Isolated Handwritten Bangla Characters using Deep Convolutional Neural Network. In *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (pp. 13-18). IEEE.

Kibria, M. R., Ahmed, A., Firdawsi, Z., & Yousuf, M. A. (2020, June). Bangla compound character recognition using support vector machine (SVM) on advanced feature sets. In *2020 IEEE Region 10 Symposium (TENSYMP)* (pp. 965-968). IEEE.

Khan, M. M., Uddin, M. S., Parvez, M. Z., & Nahar, L. (2022). A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 3356-3364.

Mukherjee, P., Sen, S., Roy, K., & Sarkar, R. (2020). Recognition of Online Handwritten Bangla Characters Using Supervised and Unsupervised Learning Approaches. *International Journal of Computer Vision and Image Processing (IJCVIP)*, *10*(3), 18-30.

Pramanik, R., & Bag, S. (2018). Shape decomposition-based handwritten compound character recognition for Bangla OCR. *Journal of Visual Communication and Image Representation*, *50*, 123-134.

Purkaystha, B., Datta, T., & Islam, M. S. (2017, December). Bengali handwritten character recognition using deep convolutional neural network. In *2017 20th International conference of computer and information technology (ICCIT)* (pp. 1-5). IEEE.

Reza, S., Amin, O. B., & Hashem, M. M. A. (2019, September). Basic to compound: a novel transfer learning approach for Bengali handwritten character recognition. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-5). IEEE.

Saha, C., Faisal, R. H., & Rahman, M. M. (2019, May). Bangla handwritten basic character recognition using deep convolutional neural network. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 190-195). IEEE.

Saha, C., Faisal, R. H., & Rahman, M. M. (2018, October). Bangla handwritten character recognition using local binary pattern and its variants. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)* (pp. 236-241). IEEE.

Sarika, N., Sirisala, N., & Velpuru, M. S. (2021, January). CNN based Optical Character Recognition and Applications. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 666-672). IEEE.

Sharif, S. M. A., Mohammed, N., Momen, S., & Mansoor, N. (2018). Classification of bangla compound characters using a hog-cnn hybrid model. In *Proceedings of the International Conference on Computing and Communication Systems* (pp. 403-411). Springer, Singapore.